

⑨ 日本国特許庁 (JP)  
⑩ 公開特許公報 (A)

⑪ 特許出願公開  
昭57—150040

⑫ Int. Cl.<sup>3</sup>  
G 06.F 9/38  
9/32

識別記号

庁内整理番号  
6745—5B  
6745—5B

⑬ 公開 昭和57年(1982)9月16日

発明の数 1  
審査請求 未請求

(全 4 頁)

⑭ パイプライン計算機

株式会社コンピュータシステム  
工場内

⑮ 特 願 昭56—34815

⑯ 出 願 人 三菱電機株式会社

⑰ 出 願 昭56(1981)3月11日

東京都千代田区丸の内2丁目2  
番3号

⑱ 発 明 者 朱雀二郎

鎌倉市上町屋325番地三菱電機

⑲ 代 理 人 弁理士 葛野信一 外1名

明 細 書

1 発明の名称

パイプライン計算機

2 特許請求の範囲

命令、データなどの先取りを司る先取りプロセッサに単一の命令ストリームに対する先取り用の命令バッファを設け、条件分岐命令解説時にこの条件分岐命令の実行を司る実行プロセッサでの実行完了時点までこの条件分岐命令が実行時に条件コードを変更する命令か否かのタグをもたせ、上記先取りプロセッサでの条件分岐命令解説時に実行直前の条件コードで分岐／非分岐が決定される条件分岐命令が解説された場合に実行中および前処理の完了した実行待ち命令中に条件コードを変更する命令がなかったときその時点の条件コードを判定することにより条件分岐命令で示される分岐先アドレスからの命令ストリームの先取り動作に移るか条件分岐命令以降の命令の前処理に移るかを決定することを特徴とするパイプライン計算機。

3 発明の詳細な説明

この発明は、命令、データの大先取りを行なうパイプライン電子計算機に関し、特に、分岐命令の分岐先命令ストリームを先取りし、命令ストリームの移行をスムーズに行うようにしたものである。

一般に、命令、データの大先取りを行なうパイプライン方式の電子計算機は第1図のような構成になっている。この第1図における1は命令やデータが格納されている主記憶装置、2は主記憶装置1の内容の一部を持つた高速バッファ記憶から成る記憶制御プロセッサ、3は記憶制御プロセッサ2から命令やデータを受けとり、命令の実行に先立つて、命令の解説、アドレス計算、オペランド準備などを司る先取りプロセッサであり、4は先取りプロセッサ3で準備された命令とオペランドデータを使つて命令を実行する実行プロセッサである。

次に動作について説明する。第2図は、大形の電子計算機で一般に採用されている先取りプロセッサ3での先取り命令の流れを示している。

特開昭57-150040 (2)

この第2図の5~8は第1図の記憶制御プロセス2から供給される命令群であり、5a~8a、5b~8b、5c~8cの3セットから成り、それぞれのセットの中では、命令は命令アドレスの昇順に準備される。

それぞれのセットは、一つが現在実行中の連続した命令ストリーム、二つ目は、現命令ストリーム中の命令解説時点で、条件分岐命令（実行される直前の条件コードで分岐／非分岐が決定される命令）があつた場合のその分岐先命令ストリーム（分岐不成功時には使用されない）、三つ目は上記分岐先命令ストリームの中にさらに条件分岐命令があつた場合のその分岐先命令ストリームがそれぞれ格納される命令バッファである。

9は、命令バッファ5~8の中から順に命令解説などのために命令ストリームを取り出し保持される命令レジスタ、10は、先取りプロセッサ3と実行プロセッサ4との実行時間のばらつきを吸収するため、準備済の命令を格納する命令キューである。

プロセスでの実行完了時点まで付けて回り、条件分岐命令が解説された場合には、実行中の命令を含む前処理済のこの命令以前の命令群のいずれも条件コードを変更しない命令の場合、その時点での条件コードによつて、分岐／非分岐を決定することによつて命令ストリームの移行をスムーズに行なうことのできるパイプライン計算機を提供することを目的とする。

以下、この発明のパイプライン計算機の実施例について説明する。第3図はその一実施例の構成を示すブロック図であり、先取りプロセッサ内のハードウェアのブロック図である。この第3図において第2図と同一部分には同一符号を付して述べる。

第3図において、5~8は1セットの命令バッファ、9は命令レジスタ、10(10a~10d)は命令キューである。11は加算器であり、オペランド・アドレス、プランタ・アドレスの算出などに使用される。12は先取りプロセッサ3の解説時点で条件分岐命令が解説され実行中および命

パイプライン計算機において、その効果を上げるためには、出現頻度の比較的多い分岐命令に対する対策が必要になる。従来の大形計算機では命令ストリームの変更をスムーズに行なうため、前述した複数の命令ストリームを持ちうる命令バッファを導入し、分岐成功／不成功が判明した時点でダイナミックに主ストリームを切替選択する方式が採られているが、その方式はハードウェアが膨大になり、制御が複雑になるばかりか、十分な容量のバッファ配憶がないと、かえつて逆効果になることもあり、中形以下のパイプライン計算機では、分岐命令に対する特別な処理はないか、あつたとしても無条件分岐命令の分岐先命令ストリームの先取り程度である。

この発明は、上記のようなハードウェア量の多い大形計算機の欠点と、分岐命令に対し殆んど無策な中形以下のパイプライン計算機の欠点を除去し、ハードウェアを殆んど追加することなく、先取りプロセッサの命令解説時点でこの命令が条件コードを変更する命令か否かのタグを実行プロセ

ッサに入っている命令がいずれも条件コードを変更しない命令の場合信号13の指示でその分岐先アドレスを保持するためのレジスタであり、14は命令ストリームを変更し、命令バッファ5~8に分岐先の新命令ストリームのフェッチを要求する信号15を発生する制御回路である。

16a~16dは命令キュー10に対応した各命令が条件コードを変更しないことを示すタグであり、16eは現在実行プロセッサで実行中の命令が条件コードを変更しないことを示すタグである。各タグ16a~16eの論理積をゲート17でとるようになつている。また、18は、現在実行中の命令の状態を示す状態レジスタでありその中に条件コードが含まれる。

次に、この発明のパイプライン計算機の動作について説明する。いま、条件分岐命令が命令レジスタ9に取り込まれたとする。命令キュー10a~10dに入っている（すべて入っているとは限らない）命令と実行中の命令とに対応するタグ16a~16eの論理積が「1」でいずれの命令も状態レジ

特開昭57-150040(3)

スタ18中の条件コードを変更しないことが、その時点で判明すると、この分岐命令の分岐条件が成立するか否かが一般に分岐命令中のフィールド定数と条件コードとの論理演算で決定され、分岐が成立すると、分岐アドレス、レジスタ12の内容とともに分岐制御回路14に伝達される。

分岐制御回路14は命令バッファ5〜8に既に先取りされているであろう旧命令ストリームをキャンセルし、分岐命令による新たな命令ストリームのフェッチ要求を出す。

一方、分岐条件が不成立の場合には、この命令以降の解説を解釈し前処理動作を続行する。

なお、上記実施例では先取りプロセッサ3の上に記憶制御プロセッサ2を想定しているが直接主記憶装置1と接続されていてもよい。また命令キューの数は特に関係ない。

この実施例では分岐条件が不成立の場合には条件分岐命令以降の解説を解釈するとしているが、必ずしも分岐命令が解説された時点で以降の命令の解説を一時中断する必要もない。

さらに、ここでは、タグによつて条件分岐以前の命令が条件コードを変更しないケースについて述べたが、変更する場合については特に規定するものではない。

ここでは条件コードを変更しないタグとしたが、条件コードを変更するタグとして論理和をとることも含むことは言うまでもない。

以上のように、この発明のパイプライン計算機によれば、先取りプロセッサに単一の命令ストリームに対する先取り用の命令バッファを設け、条件分岐命令解説時で命令の実行を司る実行プロセッサの実行完了時点まで条件分岐命令が実行時に条件コードを変更する命令か否かのタグを持たせることにより、先取りプロセッサでの命令解説時に実行直前の条件コードで分岐／非分岐が決定される条件分岐命令が解説された場合に、実行中と前処理の完了した実行待ち命令中に条件コードを変更する命令がなかつたときにその時点の条件コードを判定して条件分岐命令で示される分岐先アドレスの命令ストリームの先取り動作に移行する

か、条件分岐以降の命令の前処理に移るかを決定するようにしたので、大形機のような莫大なハードウェアと複雑な制御を必要とすることなく、条件コード不変タグと簡単なハードウェア回路によつて、条件コードが不変の場合には条件分岐命令の実行を持つことなく解説時点でそれ以降の命令前処理を決定でき、パイプライン計算機で重要な問題となる命令ストリームの乱れを大巾に改善することができる。

#### 4. 図面の簡単な説明

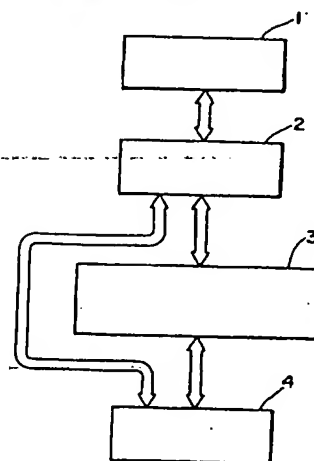
第1図はパイプライン計算機の一般的構成図、第2図は大形計算機における先取りプロセッサのブロック図、第3図はこの発明のパイプライン計算機の一実施例における先取りプロセッサ内のハードウェア・ブロック図である。

5〜8…命令バッファ、9…命令レジスタ、10…命令キュー、11…加算器、14…制御回路、16…タグ、17…ゲート、18…状態レジスタ。

なお、図中同一符号は同一または相当部分を示す。

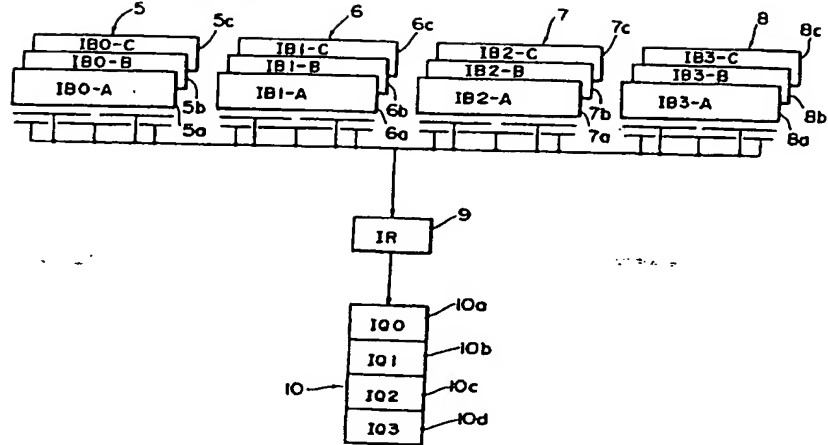
代理人 高野 信 一

第 1 図

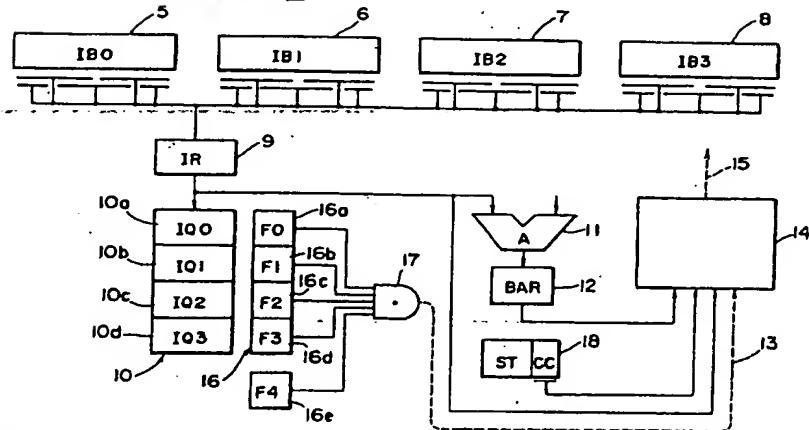


特開昭57-150040(4)

第 2 図



第 3 図



Kokai No.: S57-150040

Publication date: September 16, 1982

Application No.: S56-34815

Application date: March 11, 1981

Inventor: Suzaku Jiro

Applicant: Mitsubishi Electric Corporation

### Specification

#### [1. Title of the Invention]

Pipe-line Computer

#### [2. Claims]

[Claim 1] A pipe-line computer in which an instruction buffer for pre-fetching single instruction stream is placed in a pre-fetch processor that performs pre-fetching of an instruction and data,

wherein a tag to indicate whether the conditional branch instruction is an instruction whose condition code is changed at execution is added to a conditional branch instruction at decoding of the conditional branch instruction, until an execution processor that executes the condition branch instruction completes execution, when a conditional branch instruction which decides branch/not branch according to a condition code just before the execution is decoded at conditional branch instruction decoding with said pre-fetch processor, and if an instruction to change condition code is not found in an instruction at executing or in an execution waiting instruction whose pre-process has been completed, starting pre-fetching of an instruction stream from a branch destination address designated by the conditional branch instruction, or starting pre-processing of an instruction after the conditional branch instruction is determined by judging a condition code at that time.

#### [3. Detailed Description of the Invention]

The present invention is related to a pipe-line computer that pre-fetches especially, a branch destination instruction stream of a branch instruction is pre-fetched for shifting instruction streams smoothly.

Generally, a pipe-line type computer which performs pre-fetching of an instruction and data has a structure shown in Fig.1. In Fig.1, numeral 1 denotes a main storage device in which instructions and data are stored, numeral 2 denotes a storage control processor comprising a high speed buffer storage having a part of contents of the main storage device 1, numeral 3 denotes a pre-fetch processor which receives an instruction and data from the storage control processor 2 and performs decoding of an instruction, address calculation, operand preparation, numeral 4 denotes an execution processor to execute an instruction using an instruction prepared by the processor 3 and operand data.

Next, an operation is explained. Fig.2 shows a flow of pre-fetch instruction at the processor 3 which is generally applied to a large computer.

Numerals 5 – 8 are groups of instructions supplied by the storage control processor 2 in Fig.1, they comprise 3 sets of 5a – 8a, 5b – 8b and 5c – 8c, instructions are prepared in ascending sequence of instruction addresses in each set.

In each set, a first one is an instruction buffer for storing consecutive instruction streams which are currently being executed, a second one is an instruction buffer for storing a branch destination instruction stream when there is a conditional branch instruction (an instruction whose branch/not-branch is determined by a condition code just before the execution) in instruction decoding of the current instruction stream (the branch instruction is not used when the branch is not effected), a third one is an instruction buffer for storing an branch destination instruction stream when there is another conditional branch instruction in the branch destination stream above mentioned.

Numerals 9 denotes an instruction register for storing instruction streams for instruction decoding taken out from the instruction buffers 5 – 8 sequence, numeral 10 denotes an instruction queue to store prepared instructions for absorbing differences of execution times between the pre-fetch processor 3 and the execution processor 4.

In a pipe-line computer, in order to improve its effect, devising counter measures for branch instructions having high frequency of appearance is required. In traditional large computers, an instruction buffer which can hold said plural of instruction streams is placed for changing instruction streams smoothly, and when success/not success of branch is known, the

main stream is selectively switched dynamically. However, that method requires much amount of hardware, and then controlling becomes to be complicated, if there is not enough capacity of buffer storage, an adverse effect may appear. Therefore, in a pipe-line computer smaller than middle size, there is no special process corresponding to a branch instruction, or at most pre-fetching of a branch destination instruction stream of a non conditional branch instruction.

The present invention eliminates a defect of too much amount of hardware in a large computer and a defect of no measures for a branch instruction in a pipe-line computer smaller than middle size. The purpose of the present invention is to provide a pipe-line computer in which a tag to indicate that the instruction is an instruction that change a condition code at instruction decoding of a pre-fetch processor until completion of execution at an execution processor. When a conditional branch instruction is decoded, and if none of groups of instructions prior to the instruction including an instruction under processing is an instruction which changes a condition code, shifting of instruction streams smoothly with deciding branch/non branch according to the condition code at the moment.

Hereinafter, an embodiment of a pipe-line computer according to the present invention is explained. Fig.3 is a block diagram to show a structure of the embodiment, and a block diagram of hardware in a pre-fetch processor. In Fig.3 and Fig.2, the common parts are denoted by the same numerals.

In Fig.3, numerals 5 – 8 denote a set of instruction buffer, numeral 9 denotes an instruction register, and numeral 10 (10a – 10d) denotes an instruction queue. Numeral 11 denotes an accumulator which is used for calculation of an operand address or a branch address. Numeral 12 denotes a register for holding a branch-destination-address-according-to-an-instruction of a signal 13 when an conditional branch instruction is decoded at decoding in the pre-fetch processor 3, and no instruction under execution or in the instruction queue does not change the condition code, numeral 14 denotes a control circuit to generate a signal 15 to change instruction streams and to request fetching of a new instruction stream of the branch destination to instruction buffers 5 – 8. Numerals 16a – 16d denote tags to indicate each instructions corresponding-to-an-instruction-queue 10 does not change the condition code, and 15e denotes a tag to indicate an instruction currently under execution at the execution processor does not change the condition

code. AND of each tags 15a – 16e is obtained at gate 17. Numeral 18 denotes a state register to indicate a state of instruction currently under execution, and a condition code is included.

Next, operation of a pipe-line computer according to the present invention is explained. It is supposed that a conditional branch instruction is taken into an instruction register 9 now. When it is known that AND of instructions in instruction queues 10a – 10d (not always instructions are in all of the queues) and tags 16a – 16e corresponding to instructions under execution is "1" and any instruction changes a condition code in the state register 13, whether the branch condition of the branch instruction is effected is determined by a logical operation of a field constant and the condition code, if branch is effected, and it is notified to the branch control circuit 14 with contents in the ranch address register 12.

The branch control circuit 14 cancels old instruction streams which may have been pre-fetched in instruction buffers 5 – 8 already, and send a fetch request for a new instruction stream caused by the branch instruction.

On the other hand, when the branch condition is not effected, decoding after the instruction is canceled a ban and pre-processing operation is continued.

Though a storage control processor 2 is assumed to be on the pre-fetch processor 3 in this embodiment, the processor can be connected to a main storage device 1 directly. The number of operation queues is not directly related to the present invention.

Though decoding after the conditional branch instruction is canceled a ban when the branch condition is not effected in this embodiment, temporal abortion of instruction decoding at decoding of the branch instruction is not always necessary.

Furthermore, a case in which instructions prior to the conditional branching do not change the condition code by the tag, the case in which it is changed is not defined especially.

Though a case in which the condition code is not changed is explained here, it is needless to say that obtaining of AND with a tag which change the condition code is included.

As mentioned above, as in a pipe-line computer according to the present invention, an instruction-buffer-for-pre-fetching single instruction stream is placed in a pre-fetch processor that performs pre-fetching of an instruction and data, wherein a tag to indicate whether the conditional branch



instruction is an instruction whose condition code is changed at execution is added to a conditional branch instruction at decoding of the conditional branch instruction, until an execution processor that executes the condition branch instruction completes execution, when a conditional branch instruction which decides branch/not branch according to a condition code just before the execution is decoded at conditional branch instruction decoding with said pre-fetch processor, and if an instruction to change condition code is not found in an instruction at executing or in an execution waiting instruction whose pre-process has been completed, starting pre-fetching of an instruction stream from a branch destination address designated by the conditional branch instruction, or starting pre-processing of an instruction after the conditional branch instruction is determined by judging a condition code at that time, a large amount of hardware and complicated controlling are not necessary, when the condition code is not changed, an instruction pre-processing after decoding can be determined without waiting for execution of the conditional branch instruction with a condition code constancy tag and a simple hardware circuit, a disorder of instruction streams which is a critical problem in a pipe-line computer is largely improved.

#### [4. Brief Description of Drawings]

Fig. 1 is a drawing to show a structure of general pipe-line computer.

Fig. 2 is a block diagram to show a pre-fetch processor in a large computer.

Fig. 3 is a hardware block diagram of a pre-fetch processor in one embodiment of a pipe-line processor according to the present invention.

5 — 8 : instruction buffer

9 : instruction register

10 : instruction queue

11 : accumulator

14 : control circuit

16 : tag

17 : gate

18 : state register

In the figures, the same numerals denote the same or the equivalent parts.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**